A Coded Block Adaptive Neural Network System With a Radical-Partitioned Structure for Large-Volume Chinese Characters Recognition

MARK W. MAO AND JAMES B. KUO
National Taiwan University

Abstract—This paper presents a coded block adaptive neural network system using a radical-partitioned structure for large-volume Chinese characters recognition. Using the coded block adaptive neural network system with a radical-partitioned structure, 1,000 frequently-used Chinese characters have been successfully trained in 139.2 hours using an 18 MIPs computer. According to the simulation results, the coded block system with a radical-partitioned structure provides an acceptable learning time, a good recognition rate, and an excellent expansion capability for large-volume Chinese characters recognition.

Keywords—Neural network, Radical, Chinese characters, Recognition, Coded, Block, Adaptive.

1. INTRODUCTION

A pattern recognition system, as shown in Figure 1, is composed of a translation-invariant network and a standard adaptive two-layer network. The translation-invariant network is used to map a retinal image into multibit outputs (Rumelhart & McClelland, 1986; Widrow, Winter & Baxter, 1988). The standard adaptive two-layer network (Mirchandani & Cao, 1989) made of \(2n^2\) adaptive neurons and \(2n^4\) connections between them, can be trained to provide output responses corresponding to the original image as required. Software implementation of the adaptive neural network using back propagation (Vogl, Mangis, & Rigler, 1988; Yamada, Kami, Tsukumo, & Temma, 1989) has been demonstrated. A structured neural network architecture to speed up learning for hand written digit recognition has been reported (Le Cun et al., 1989). In fact, for Western characters recognition, only about 36 alphanumerics are needed. However, for Chinese characters recognition, over 20,000 characters are necessary. For large-volume Chinese characters recognition, learning speed is determined by the number of neurons and connections between them. How to train large-volume Chinese characters using an efficient neural network structure has been a challenging effort. Recently, a block adaptive network structure using \(n^2\) local blocks for an \(n \times n\) input array as shown in Figure 2 has been used to enhance the learning speed (Chen, Hsiao, & Kuo, 1990; Kuo, Chen, & Hsiao, in press) for alphanumerics recognition. However, the block adaptive network structure requires \(n^2\) local blocks, which is still a bottleneck for training large-volume Chinese characters. A coded block neural network system, which requires an order-of-magnitude fewer local blocks, has been used to provide a faster learning process for the alphanumerics recognition (Kuo et al., 1991; Kuo & Mao, 1991). In this paper a coded block adaptive neural network system with a radical-partitioned structure for large-volume Chinese characters recognition is presented. In the following sections, the radical-partitioned coded block structure is introduced, followed by the algorithm used in the system and simulation results.

2. THE CODED BLOCK ADAPTIVE RADICAL-PARTITIONED STRUCTURE

Figure 2 (top) shows the block adaptive network structure for an \(n \times n\) input array (Chen et al., 1990; Kuo et al., in press). In the block structure, the system is partitioned into \(n^2\) blocks. Each block, as shown in Figure 2 (bottom), contains \(S \pm 1\) neurons and \(S n^2 + S\) connections between them, which are grouped into two layers. Instead of \(2n^2\) global connections in two layers of the standard structure, only \(S n^2\) local connec-
Instead of \( n^2 \) local blocks, only less than \( n \) local blocks with outputs coded corresponding to the input patterns are used in the system. Using the coded block system, the training time can be shortened further owing to the drastic reduction in the number of local blocks. The coded local block system is suitable for training the system with small-volume Chinese characters. However, for training over 200 Chinese characters, the coded block adaptive neural network using back propagation algorithm suffers from convergence difficulties.
Although there are over 20,000 Chinese characters, they can be categorized into about 230 radical groups. Each radical group has 5–200 characters containing an identical radical portion as shown in Figure 4a. Taking advantage of the radical properties, the coded block adaptive neural network system can be reconfigured as shown in Figure 4b such that each coded block is assigned to be responsible for training Chinese characters having the same radical portion. Using a radical-partitioned structure, as shown in Figure 4b, the coded block adaptive neural network can overcome the convergence difficulties during training since training of the large-volume Chinese characters can be divided into small procedures associated with radical blocks. In fact, the training of each radical block can be divided further into steps associated with internal local blocks. Therefore, using the coded block system with a radical-partitioned structure, the overall training procedure is simplified from two approaches. First, Chinese characters as input training patterns are categorized into several radical groups. Chinese characters in each radical group having an identical radical portion is assigned to a radical block. Second, the training of each radical block is divided into steps associated with several internal local blocks. Training of each internal local block
is independent. Therefore, convergence during training is determined by a very small portion of the overall system—the internal local block in each radical block with a small portion of the Chinese characters to train. Therefore, convergence is guaranteed and training time is much shorter.

As shown in Figure 4b, above the coded radical blocks, there is a radical selector, which is used to identify the radical category of each input Chinese character during recognition. As shown in Figure 4c, during recognition in the radical selector block, dot products of each input Chinese character vector \( \bar{T} \) and all radical vectors \( \bar{R} \) are carried out, followed by normalization by the norm of the radical vector \( \bar{R} \). Then, according to the highest value of the normalized dot products, a decision is made on the input character about which radical group it belongs to. Then, the input character is transferred to the associated radical block for further processing. Since only one radical block handles the second-step processing during recognition, the recognition time is greatly shortened as compared to the coded block adaptive neural network system without the radical-partitioned structure. Consequently, the coded block adaptive neural network system with the radical-partitioned structure can be advantageous for both training and recognition of large-volume Chinese characters.

3. THE BACK PROPAGATION ALGORITHM

Back propagation algorithm (Jones & Hoskins, 1987; Vogl et al., 1988; Yamada et al., 1989) has been used in the coded block adaptive neural network structure. The objective of the back propagation algorithm is to establish the desired responses for the neurons by propagating back the output error signal through the network to modify the weight values. First, present a simple pattern to the inputs of the network primed with random initial weights. Change the second layer weights \( w_{jk} \) according to:

\[
\Delta w_{jk}(n) = \eta \delta_j h_k + \alpha \Delta w_{jk}(n-1),
\]

where \( \eta \) is the learning rate, \( \alpha \) is the smoothing factor, the \( \delta_j \) is defined as \( \delta_j = \mu(1 - u_j)(t_j - u_j) \) with the second layer, and desired and actual outputs \( t_j, u_j \) and \( h_k \) will be defined below. After computing \( \delta_j \) in the output layer, the first layer values \( \delta_k^o \) can be obtained:

\[
\delta_k^o = h_k(1 - h_k) \sum_j \delta_j w_{kj},
\]

where

\[
h_k = \frac{1}{1 + e^{-\sum_j w_{jk} s_j}}.
\]

The weight \( w_{jk} \) associated with the synapse between input \( s_j \) and the first layer \( k \)-th neuron is:

\[
\Delta w_{jk}(n) = \eta \delta_k^o s_j + \alpha \Delta w_{jk}(n-1).
\]

Several training epochs of pattern activation and error feedback are performed with all the training patterns until the output of each local block for all input patterns is correct.

Applying the back propagation algorithm, the adaptive neural network with the local block structure has advantages in learning time. The intrinsic better properties of the local block system can be reasoned as follows. Assume that \( x_1, \ldots, x_n \) are \( n \) dimensional input patterns and \( y_1, \ldots, y_n \) are corresponding outputs. For a standard neural network structure without local blocks, after training, a common function, \( f \), representing the trained weights can be obtained such that:

\[
f: x_1 \rightarrow y_1 \\
f: x_2 \rightarrow y_2 \\
\vdots
\]

On the other hand, in the system with the local block structure, a simpler common function, \( g \), can be identified for each block.

\[
g: x_1 \rightarrow (1 \text{ or } 0) \\
g: x_2 \rightarrow (1 \text{ or } 0) \\
\vdots
\]

Based on the above analysis, obviously a common function, \( g \), can be obtained much more easily for the local block systems. Therefore, the local block structure can have a quicker convergence during training.

4. SIMULATION RESULTS AND DISCUSSION

In order to investigate the potential of the coded block adaptive neural network system with a radical-partitioned structure for large-volume Chinese characters recognition, the performance in terms of learning time, reconfiguration based on simulation results has been obtained. One thousand Chinese characters composed of \( 24 \times 24 \) lattice dots in 25 frequently-used radical groups as shown in Figure 5 have been used as the training patterns for the coded block adaptive neural network system with a radical-partitioned structure. Each radical block is trained with its 40 associated Chinese characters independently. Each of the radical blocks contains 16 internal local blocks. In the coded block adaptive neural network system with a radical-partitioned structure, a learning rate, \( \eta \), which affects the learning speed of a block adaptive neural network system using back-propagation algorithm (Chen et al., 1990; Kuo et al., in press), of 0.2 has been used. In addition, the number of first layer neurons in the local block structure, which also affects the learning time, is defined as 12.
FIGURE 5. The 1,000 Chinese characters used as the training patterns.

Owing to the radical-partitioned structure used in the coded block neural network system, the training process can be divided into steps associated with the 400 internal local blocks in 25 radical blocks. Training each of the 400 internal local blocks is independent from one another. Table 1 shows the epoches needed for training each of 400 internal local blocks with 25 sets of Chinese characters. Each epoch of training each local block with 40 corresponding Chinese characters once took about 20 sec CPU time on an 18-MIPs computer. Each local block may have different number of epoches needed to accomplish training, which depends on the input patterns and the initial synaptic weights. Some internal local blocks do not need training at all as indicated as “0” in Table 1. Figure 6 shows the learning curves in terms of average number of epoches needed for training each local block in all 25 radical blocks with its corresponding 40 Chinese characters. Among 16 internal local blocks, the training time varies. On the average, each local block in 25 radical blocks reaches a convergence within 100 epoches during training. The total learning time for training the 25 radical block system with 1,000 Chinese characters is 139.2 hours CPU time on an 18 MIPs computer. Without the radical-partitioned coded block neural network system, the training time of 1,000 Chinese characters is just not feasible.

Neural networks have renown properties in reconfigurability and fault-tolerance, which are important for a reliable pattern recognition system (Denker & Wittner, 1987; Feldman & Ballard, 1982). In addition to an acceptable learning time, the local block structure has good properties in reconfigurability and expansion capability. Figure 7 shows the average output percentage error associated with the radical-partitioned coded block system using random input samples taken from several radicals. Each input sample is with random errors. The total bits of random errors out of the 24 x 24 lattice dots in each input sample are expressed as the percentage random error. Recognizing a character out of 1,000 Chinese characters as shown in Figure 5 took about less than 0.1 sec CPU time on an 18 MIPs computer. The case with 40 random samples with percentage random error taken from two radicals indicates the highest recognition rate. The cases with more random samples with percentage random error taken from more radical sections show a worse recognition rate. The cases with over 400 random samples seem to have a near constant recognition rate of over 80% regardless of the number of samples used. The overall recognition rate exceeds 75% regardless of the input percentage error. In fact, the recognition rate is mainly determined by the radical selector since the recognition rate of all local radical blocks for all the cases under study is almost perfect. The recognition rate can be enhanced if preprocessing of the input characters before the selector block can be done.
<table>
<thead>
<tr>
<th>Rad</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rad1</td>
<td>0</td>
<td>35</td>
<td>77</td>
<td>84</td>
<td>26</td>
<td>65</td>
<td>25</td>
<td>112</td>
<td>80</td>
<td>14</td>
<td>25</td>
<td>27</td>
<td>35</td>
<td>35</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>Rad2</td>
<td>0</td>
<td>23</td>
<td>28</td>
<td>32</td>
<td>43</td>
<td>55</td>
<td>23</td>
<td>30</td>
<td>37</td>
<td>18</td>
<td>40</td>
<td>70</td>
<td>11</td>
<td>23</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Rad3</td>
<td>0</td>
<td>121</td>
<td>21</td>
<td>33</td>
<td>36</td>
<td>27</td>
<td>52</td>
<td>28</td>
<td>21</td>
<td>36</td>
<td>33</td>
<td>28</td>
<td>44</td>
<td>89</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Rad4</td>
<td>0</td>
<td>63</td>
<td>155</td>
<td>142</td>
<td>77</td>
<td>84</td>
<td>26</td>
<td>65</td>
<td>25</td>
<td>112</td>
<td>80</td>
<td>14</td>
<td>25</td>
<td>31</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>Rad5</td>
<td>0</td>
<td>198</td>
<td>24</td>
<td>24</td>
<td>50</td>
<td>62</td>
<td>25</td>
<td>102</td>
<td>72</td>
<td>44</td>
<td>80</td>
<td>61</td>
<td>18</td>
<td>47</td>
<td>118</td>
<td></td>
</tr>
<tr>
<td>Rad6</td>
<td>0</td>
<td>475</td>
<td>168</td>
<td>113</td>
<td>85</td>
<td>35</td>
<td>24</td>
<td>116</td>
<td>77</td>
<td>259</td>
<td>24</td>
<td>81</td>
<td>36</td>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad7</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>42</td>
<td>24</td>
<td>15</td>
<td>53</td>
<td>29</td>
<td>13</td>
<td>31</td>
<td>51</td>
<td>14</td>
<td>102</td>
<td>31</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>Rad8</td>
<td>0</td>
<td>38</td>
<td>33</td>
<td>45</td>
<td>43</td>
<td>210</td>
<td>258</td>
<td>176</td>
<td>24</td>
<td>45</td>
<td>23</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad9</td>
<td>0</td>
<td>66</td>
<td>34</td>
<td>202</td>
<td>28</td>
<td>161</td>
<td>91</td>
<td>250</td>
<td>50</td>
<td>28</td>
<td>83</td>
<td>49</td>
<td>99</td>
<td>36</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Rad10</td>
<td>0</td>
<td>21</td>
<td>27</td>
<td>133</td>
<td>28</td>
<td>162</td>
<td>79</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td>121</td>
<td>29</td>
<td>25</td>
<td>91</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Rad11</td>
<td>0</td>
<td>60</td>
<td>63</td>
<td>87</td>
<td>186</td>
<td>25</td>
<td>132</td>
<td>106</td>
<td>148</td>
<td>18</td>
<td>20</td>
<td>32</td>
<td>9</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad12</td>
<td>0</td>
<td>132</td>
<td>193</td>
<td>187</td>
<td>73</td>
<td>285</td>
<td>101</td>
<td>258</td>
<td>40</td>
<td>56</td>
<td>32</td>
<td>40</td>
<td>31</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad13</td>
<td>0</td>
<td>16</td>
<td>20</td>
<td>20</td>
<td>66</td>
<td>73</td>
<td>53</td>
<td>31</td>
<td>99</td>
<td>0</td>
<td>52</td>
<td>18</td>
<td>41</td>
<td>39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad14</td>
<td>0</td>
<td>81</td>
<td>52</td>
<td>154</td>
<td>34</td>
<td>28</td>
<td>202</td>
<td>69</td>
<td>57</td>
<td>30</td>
<td>20</td>
<td>18</td>
<td>11</td>
<td>74</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad15</td>
<td>0</td>
<td>0</td>
<td>31</td>
<td>23</td>
<td>29</td>
<td>16</td>
<td>90</td>
<td>21</td>
<td>60</td>
<td>38</td>
<td>37</td>
<td>19</td>
<td>28</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad16</td>
<td>0</td>
<td>11</td>
<td>8</td>
<td>11</td>
<td>12</td>
<td>119</td>
<td>49</td>
<td>44</td>
<td>73</td>
<td>14</td>
<td>39</td>
<td>60</td>
<td>14</td>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad17</td>
<td>0</td>
<td>0</td>
<td>106</td>
<td>150</td>
<td>38</td>
<td>31</td>
<td>23</td>
<td>42</td>
<td>75</td>
<td>44</td>
<td>55</td>
<td>36</td>
<td>70</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad18</td>
<td>0</td>
<td>31</td>
<td>430</td>
<td>110</td>
<td>93</td>
<td>52</td>
<td>67</td>
<td>31</td>
<td>41</td>
<td>40</td>
<td>33</td>
<td>160</td>
<td>50</td>
<td>72</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad19</td>
<td>0</td>
<td>60</td>
<td>60</td>
<td>33</td>
<td>74</td>
<td>81</td>
<td>26</td>
<td>19</td>
<td>147</td>
<td>60</td>
<td>48</td>
<td>64</td>
<td>35</td>
<td>160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad20</td>
<td>0</td>
<td>0</td>
<td>70</td>
<td>34</td>
<td>432</td>
<td>109</td>
<td>97</td>
<td>55</td>
<td>34</td>
<td>117</td>
<td>48</td>
<td>64</td>
<td>16</td>
<td>180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad21</td>
<td>0</td>
<td>26</td>
<td>40</td>
<td>112</td>
<td>31</td>
<td>174</td>
<td>26</td>
<td>102</td>
<td>32</td>
<td>43</td>
<td>51</td>
<td>39</td>
<td>79</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad22</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>155</td>
<td>22</td>
<td>15</td>
<td>16</td>
<td>19</td>
<td>19</td>
<td>13</td>
<td>11</td>
<td>26</td>
<td>64</td>
<td>144</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad23</td>
<td>0</td>
<td>44</td>
<td>72</td>
<td>71</td>
<td>40</td>
<td>83</td>
<td>16</td>
<td>19</td>
<td>19</td>
<td>13</td>
<td>11</td>
<td>26</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad24</td>
<td>0</td>
<td>86</td>
<td>90</td>
<td>26</td>
<td>42</td>
<td>26</td>
<td>58</td>
<td>42</td>
<td>132</td>
<td>20</td>
<td>155</td>
<td>165</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rad25</td>
<td>0</td>
<td>43</td>
<td>48</td>
<td>81</td>
<td>403</td>
<td>139</td>
<td>25</td>
<td>30</td>
<td>41</td>
<td>92</td>
<td>24</td>
<td>115</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ave.</td>
<td>0</td>
<td>65</td>
<td>77</td>
<td>76</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Numbers of Epoches Needed for Training Local Blocks in all Radical Blocks

Consequently, in building a large Chinese characters system, the training process does not have to start from the very beginning every time. Instead, the previous trained radical blocks, which relate to the Chinese characters of identical radicals, can be used. Only the new untrained characters need to be trained. Consequently, the cost in training can be greatly reduced. With the radical-partitioned structure, radical coded blocks can

![Figure 6](image-url)
be concatenated to build a large coded block adaptive neural network system for large-volume Chinese characters recognition. In addition, the convergence of training the growing system is always guaranteed since each time only the newly-added characters belonging to the same radical group need to be trained. Furthermore, taking advantage of the expansion capability of the system, recognition of multifont printed Chinese characters or handwritten Chinese characters can be expedited via adding extra local radical blocks of various printed fonts and various possible handwritten styles.

5. CONCLUSION

In this paper, an efficient coded block adaptive neural network system with a radical-partitioned structure suitable for large-volume Chinese characters recognition, has been successfully trained with 1,000 Chinese characters in 139.2 hours CPU time on a 18 MIPS computer. According to simulation results, the coded block adaptive network with a radical-partitioned structure provides an acceptable learning time, a good reconfigurability, and an excellent expansion capability.

REFERENCES


